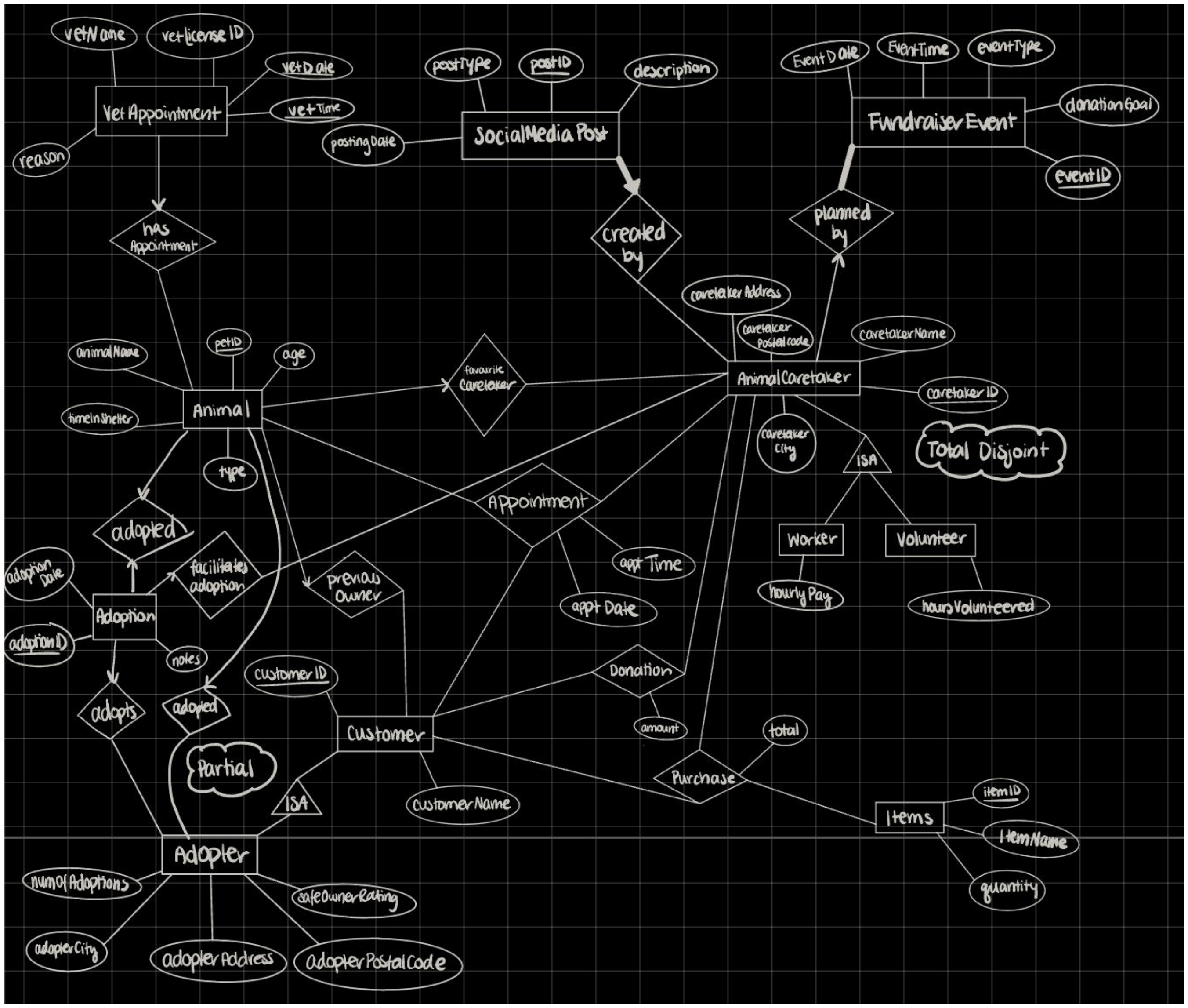
Milestone 2

1. Project Summary

Our project is a pet shelter management application. The goal is to allow application users to efficiently manage all areas of a pet shelter, including the adoption and care of animals, appointments, staff management, social media, event planning, and inventory management. This application will utilize the Oracle database management system to manage relationships between animals, adopters, employees and volunteers and facilitate a smooth and efficient adoption experience for everyone involved.

2. Updated ER Diagram



Changes from the ER Diagram submitted in Milestone 1:

- Adopter: new entity, used to describe person who adopted the animal
    - Partial ISA relationship with Customer (not all Customers are Adopters)
    - numberofAdoptions: describing the number of previous adoptions they had at this animal shelter
    - safeOwnerRating: describing the owner's reliability on a scale of 1 to 5, determined by the employee
- Adoption: new entity with attributes: adoptionID, date of the adoption, notes about the adoption, animal that was adopted, caretaker that facilitated the adoption, adopter than adopted the animal
- Added relationship between Animal and Adopter for storing the adopterID for an adopted animal (NULL if not adopted)
- Booked Appointment -> Vet Appointment : changed the name of the relationship to make it clearer
- Removed Veterinarian entity and included attributes in VetAppointment entity
- Changed PK of VetAppointment to be date and time (to function like a calendar)
- Type: attribute of Animal (like cat, dog, etc.)
- hoursVolunteered: attribute of the Volunteer entity, describing the hours volunteered
- hourlyPay: attribute of the Worker entity, describing the hourly pay of the worker
- Clarified that the ISA relationship between AnimalCaretaker and Worker, Volunteer is total disjoint
- favorite -> favoriteCaretaker: changed the name of the relationship between Animal and Animal Caretaker for clarity
- Included total participation for SocialMediaPost to have an AnimalCaretaker that created it
- Included total participation for FundraiserEvent to always have at least one AnimalCaretaker have that eventID as what they are planning
- Added address information to Adopter and AnimalCaretaker entities
- Made some generic attribute names more specific (i.e. type -> eventType, name -> itemName, name -> vetName, etc.)

Changes from the ER Diagram submitted in Milestone 1:

- Removed participation of Animal in Purchase relationship as it was a typo (relational schema from milestone 2 was correct)

3. Schema

Animal(petID: integer, animalName: varchar, type: varchar, age: integer, **favouriteCaretaker**: integer, **previousOwner**: integer, timeInShelter: integer, **adopterID**: integer)

- **Foreign key:** favoriteCaretaker is a foreign key to caretakerID in AnimalCaretaker
- **Foreign key:** previousOwner is a foreign key to customerID in Customer
- **Foreign key:** adopterID is a foreign key to adopterID in Adopter
- **Additional notes:** timeInShelter is in weeks, and adopterID is null if animal not yet adopted

VetAppointment(vetDate: date, vetTime: time, vetLicenseID: integer, vetName: varchar, reason: varchar, **petID**: integer)
- **Foreign key:** petID is foreign key for Animal petID who the appointment is booked for

AnimalCaretaker(caretakerID: integer, caretakerName: varchar, **fundEventID**: integer, caretakerAddress: varchar, caretakerPostalCode: varchar, caretakerCity: varchar)
- **Foreign key:** fundEventID is foreign key for FundraiserEvent eventID they are planning

Worker(**workerID**: integer, hourlyPay: integer)
- **Foreign key:** workerID is foreign key for AnimalCaretaker caretakerID

Volunteer(**volunteerID**: integer, hoursVolunteered: integer)
- **Foreign key:** volunteerID is foreign key for AnimalCaretaker caretakerID

FundraiserEvent(eventID: integer, eventType: varchar, eventDate: date, eventTime: time, donationGoal: integer), eventDate is unique
- **Additional notes:** For every instance of FundraiserEvent, some AnimalCaretaker must have its eventID as a fundEventID, eventDate is unique since you shouldn't have more than one event on any given day.

SocialMediaPost(postID: integer, postType: varchar, description: varchar, postingDate: date **caretakerID**: integer), caretakerID not null, postingDate is unique
- **Foreign key:** creatorID is foreign key to animalCaretaker caretakerID
- **Additional notes:** postingDate is unique since you don't want to spam your follower's feeds

Customer(customerID: integer, customerName: varchar)

Adopter(**adopterID**: integer, numOfAdoptions: integer, safeOwnerRating: integer, adopterPostalCode: varchar, adopterCity: varchar, adopterAddress: varchar)
- **Foreign key:** adopterID is foreign key for customerID in Customer
- **Additional notes:** Safe owner rating is on a scale of 1 to 5, determined by an employee

Adoption(adoptionID: integer, **petID**: integer, **adopterID**: integer, **caretakerID**: integer, adoptionDate: date, notes: varchar)

- **Foreign key:** petID is foreign key to petID in Animal
- **Foreign key:** adopterID is foreign key to adopterID in Adopter
- **Foreign key:** caretakerID is foreign key to caretakerID in AnimalCaretaker that facilitated adoption
- PetID is unique since each adoption pertains to only one animal

Appointment(**petID**: integer, **caretakerID**: integer, **customerID**: integer, apptDate: date, apptTime: time)

- **Foreign key:** petID is foreign key to petID in Animal
- **Foreign key:** caretakerID is foreign key to caretakerID in AnimalCaretaker that facilitates appointment
- **Foreign key:** customerID is foreign key to customerID in Customer

Donation(**customerID**: integer, **caretakerID**: integer, amount: integer)

- **Foreign key:** customerID is foreign key to customerID in Customer
- **Foreign key:** caretakerID is foreign key to caretakerID in AnimalCaretaker that facilitates donation

Item(itemID: integer, itemName: varchar, quantity: integer)

Purchase(**customerID**: integer, **caretakerID**: integer, **itemID**: integer, total: integer)

- **Foreign key:** customerID is foreign key to customerID in Customer
- **Foreign key:** caretakerID is a foreign key to caretakerID in AnimalCaretaker
- **Foreign key:** itemID is a foreign key to itemID in Items

4. Functional Dependencies

Animal:
petID → animalName, age, timeInShelter, type, favouriteCaretaker, previousOwner, adopterID

VetAppointment:
petID → vetLicenseID, vetName, vetDate, vetTime, reason
vetLicenseID → vetName

AnimalCaretaker:
caretakerID ➜ caretakerName, fundEventID, caretakerAddress, caretakerPostalCode, caretakerCity
caretakerPostalCode ➜ caretakerCity

Worker:
workerID ➜ caretakerID, hourlyPay

Volunteer:
volunteerID ➜ caretakerID, hoursVolunteered

FundraiserEvent:
eventID ➜ eventDate, eventTime, eventType, donationGoal
eventDate ➜ eventTime

SocialMediaPost:
postID ➜ postType, description, caretakerID, postingDate
postingDate ➜ postType

Customer:
customerID ➜ customerName

Adopter:
customerID ➜ numOfAdoptions, safeOwnerRating, adopterAddress, adopterPostalCode, adopterCity
adopterPostalCode ➜ adopterCity

Adoption:
adoptionID ➜ petID, adopterID, caretakerID, adoptionDate, notes
petID ➜ adopterID

Appointment:
petID, caretakerID, customerID ➜ apptDate, apptTime

Donation:
customerID, caretakerID ➜ amount

Item:
itemID ➜ itemName, quantity
itemName ➜ itemID

Purchase:
customerID, caretakerID, itemID ➝ total
itemID ➝ total

5. <u>Normalization</u>

5.1 Animal

Relation:
Animal(<u>petID</u>: integer, animalName: varchar, type: varchar, age: integer, **favouriteCaretaker**: integer, **previousOwner**: integer, timeInShelter: integer, **adopterID**: integer)
Functional dependencies:
petID ➝ animalName, age, timeInShelter, type, favouriteCaretaker, previousOwner, adopterID

Is this in BCNF?
Since the only FD is the primary key determining the other attributes, this relation is in BCNF already.

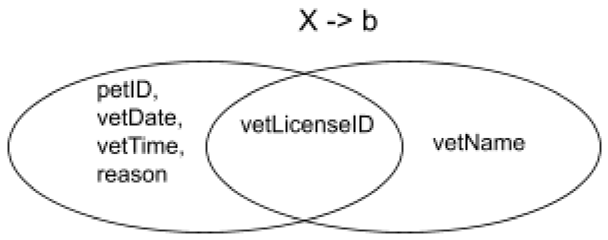
5.2 VetAppointment

Relation:
VetAppointment(<u>vetDate</u>: date, <u>vetTime</u>: time, vetLicenseID: integer, vetName: varchar, reason: varchar, **petID**: integer)
Functional dependencies:
petID ➝ vetLicenseID, vetName, vetDate, vetTime, reason
vetLicenseID ➝ vetName

Is this in BCNF?
Take closures:
petID+ = {petID, vetLicenseID, vetName, vetDate, vetTime, reason}
vetLicenseID+ = {vetLicenseID, vetName}

- petID is a superkey, so is in BCNF
- vetLicenseID is not a superkey ➝ violates BCNF

Decompose on *vetLicenseID ➝ vetName*



VetAppointment(petID, vetDate, vetTime, reason, vetLicenseID)
Vet(vetLicenseID, vetName)

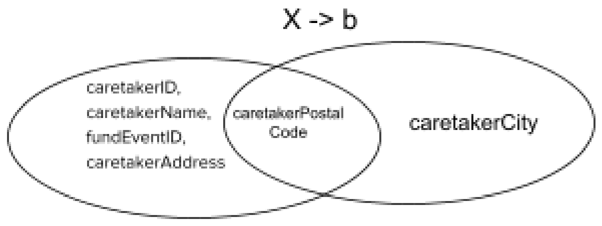
5.3 AnimalCaretaker

Relation:
AnimalCaretaker(<u>caretakerID</u>: integer, caretakerName: varchar, **fundEventID**: integer, caretakerAddress: varchar, caretakerPostalCode: varchar, caretakerCity: varchar)
Functional dependencies:
caretakerID ➝ caretakerName, fundEventID, caretakerAddress, caretakerPostalCode, caretakerCity
caretakerPostalCode ➝ caretakerCity

Is this in BCNF?
Take closures:
caretakerID+ = {caretakerID, caretakerName, fundEventID, caretakerAddress, caretakerPostalCode, caretakerCity}
caretakerPostalCode+ = {caretakerPostalCode, caretakerCity}

- caretakerID is a superkey, so is in BCNF
- However looking at the second FD caretakerPostalCode is not a superkey, so this violates BCNF

Decompose on second FD



AnimalCaretakers(caretakerID, caretakerName, fundEventID, caretakerAddress, caretakerPostalCode)
AnimalCaretakerPC(caretakerPostalCode, caretakerCity)

5.4 Worker

Relation:

Worker(**workerID**: integer, hourlyPay: integer)
Functional dependencies:
workerID ➔ caretakerID, hourlyPay

Is this in BCNF?
Since the only FD is the primary key determining the other attributes, this relation is in BCNF already.

5.5 Volunteer

Relation: Volunteer(**volunteerID**: integer, hoursVolunteered: integer)
Functional dependencies:
volunteerID ➔ caretakerID, hoursVolunteered

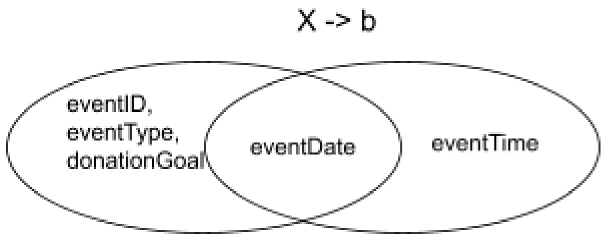
Is this in BCNF?
Since the only FD is the primary key determining the other attributes, this relation is in BCNF already.

5.6 FundraiserEvent

Relation:
FundraiserEvent(eventID: integer, eventType: varchar, eventDate: date, eventTime: time, donationGoal: integer)
Functional dependencies:
eventID ➔ eventDate, eventTime, eventType, donationGoal
eventDate ➔ eventTime

- eventID is a superkey, so is in BCNF
- eventDate is not a superkey ➔ second FD violates BCNF

Decompose on second FD



FundraiserEvent(eventID, eventType, donationGoal, eventTime, eventDate)
FundraiserEventDate(eventDate, eventTime)

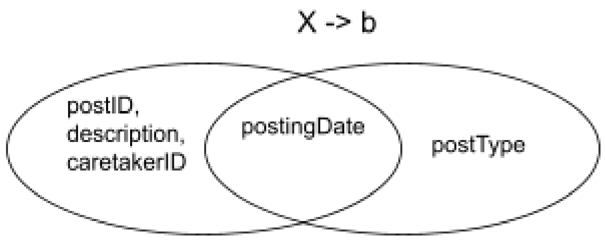
5.7 SocialMediaPost

Relation:

SocialMediaPost(postID: integer, postType: varchar, description: varchar, postingDate: date **caretakerID**: integer)
Functional dependencies:
postID → postType, description, caretakerID, postingDate
postingDate → postType

Is this in BCNF?
Take closures
postID+ = {postID, postType, description, caretakerID, postingDate}
postingDate+ = {postingDate, postType}
- postID is a superkey, so is in BCNF
- However looking at the second FD postingDate is not a superkey, so this violates BCNF

Decompose on second FD



Post(postID, description, caretakerID, postingDate)
PostDateAndType(postingDate, postType)

5.8 Customer
Relation:
Customer(customerID: integer, customerName: varchar)
Functional dependencies:
customerID → customerName

Is this in BCNF?
Since the only FD is the primary key determining the other attributes, this relation is in BCNF already.

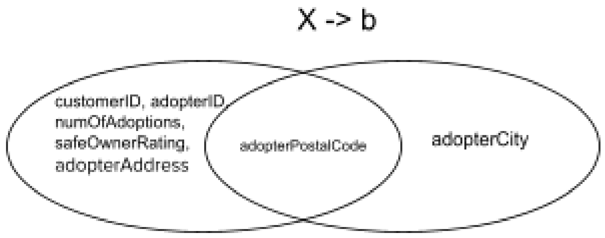
5.9 Adopter
Relation:
Adopter(**adopterID**: integer, numOfAdoptions: integer, safeOwnerRating: integer, adopterPostalCode: varchar, adopterCity: varchar, adopterAddress: varchar)
Functional dependencies:

customerID ➔ numOfAdoptions, safeOwnerRating, adopterAddress, adopterPostalCode, adopterCity
adopterPostalCode ➔ adopterCity

Is this in BCNF?
Take closures
adopterID+ = { adopterID, numOfAdoptions, safeOwnerRating, adopterPostalCode, adopterCity, adopterAddress}
adopterPostalCode+ = {adopterPostalCode, adopterCity}

- adopterID is a superkey, so is in BCNF
- adopterPostalCode is not a superkey ➔ it violates BCNF

Decompose on second FD


X -> b
customerID, adopterID, numOfAdoptions, safeOwnerRating, adopterAddress
adopterPostalCode
adopterCity

Adopter(adopterID, numOfAdoptions, safeOwnerRating, adopterPostalCode, adopterAddress)
AdopterPCs(adopterPostalCode, adopterCity)

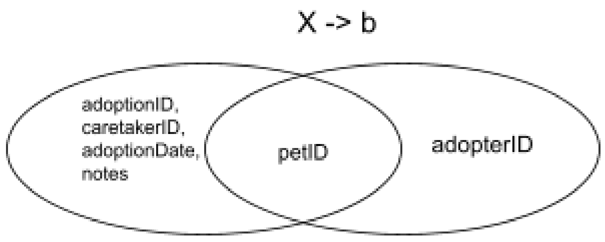
5.10 Adoption

Relation:
Adoption(<u>adoptionID</u>: integer, **petID**: integer, **adopterID**: integer, **caretakerID**: integer, adoptionDate: date, notes: varchar)
Functional dependencies:
adoptionID ➔ petID, adopterID, caretakerID, adoptionDate, notes
petID ➔ adopterID

Is this in BCNF?
Take closures:
adoptionID+ = {adoptionID, petID, adopterID, caretakerID, adoptionDate, notes}
petID+ = {petID, adopterID}

- adoptionID is a superkey, so it is in BCNF
- petID is not a superkey ➔ it violates BCNF

Decompose on second FD



AdoptionDetails(adoptionID, petID, caretakerID, adoptionDate, notes)
PetAdopter(petID, adopterID)

5.11 Appointment

Relation:
Appointment(<u>**petID**</u>: integer, <u>**caretakerID**</u>: integer, <u>**customerID**</u>: integer, apptDate: date, apptTime: time)
Functional dependencies:
petID, caretakerID, customerID ➜ apptDate, apptTime

Is this in BCNF?
Since the only FD is the primary key determining the other attributes, this relation is in BCNF already.

5.12 Donation

Relation:
Donation(<u>**customerID**</u>: integer, <u>**caretakerID**</u>: integer, amount: integer)
Functional dependencies:
customerID, caretakerID ➜ amount

Is this in BCNF?
Since the only FD is the primary key determining the other attributes, this relation is in BCNF already.

5.13 Item

Relation:
Item(<u>itemID</u>, itemName, quantity)
Functional dependencies:
itemID ➜ itemName, quantity
itemName ➜ itemID

Is this in BCNF?
Each functional dependency is in BCNF since itemName determines itemID, which is a superkey.

5.14 Purchase

Relation:

Purchase(**<u>customerID</u>**: integer, **<u>caretakerID</u>**: integer, **<u>itemID</u>**: integer, total: integer)

Functional Dependencies:

customerID, caretakerID, itemID ➔ total

itemID ➔ total

Is this in BCNF?

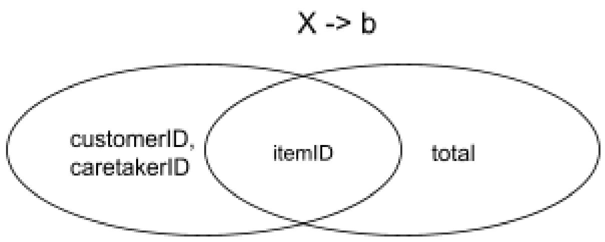
Take closures

customerID, caretakerID, itemID+ = {customerID, caretakerID, itemID, total}

itemID+ = {itemID, total}

- {customerID, caretakerID, itemID} is a superkey, so is in BCNF
- However looking at the second FD itemID by itself is not a superkey, so this violates BCNF

Decompose on second FD



ItemPrice(itemID,total)

ItemPurchase(ItemID, customerID, caretakerID)

After Normalization

Animal(<u>petID</u>: integer, animalName: varchar, type: varchar, age: integer, **favouriteCaretaker**: integer, **previousOwner**: integer, timeInShelter: integer, adopterID: integer)

- **Foreign key:** favoriteCaretaker is a foreign key to caretakerID in AnimalCaretaker
- **Foreign key:** previousOwner is a foreign key to customerID in Customer
- **Foreign key:** adopterID is a foreign key to adopterID in Adopter
- **Additional notes:** timeInShelter is in weeks, and adopterID is null if animal not yet adopted

VetAppointment(<u>vetDate</u>: date, <u>vetTime</u>: time, vetLicenseID: integer, reason: varchar, **petID**: integer)

- **Foreign key:** petID is foreign key for Animal petID who the appointment is booked for

Vet(vetLicenseID: integer, vetName: varchar)

AnimalCaretakers(caretakerID: integer, caretakerName: varchar, **fundEventID**: integer, caretakerAddress: varchar, caretakerPostalCode: varchar)
- **Foreign key:** fundEventID is foreign key for FundraiserEvent eventID they are planning

AnimalCaretakerPC(caretakerPostalCode: varchar, caretakerCity: varchar)

Worker(**workerID**: integer, hourlyPay: integer)
- **Foreign key:** workerID is foreign key for AnimalCaretaker caretakerID

Volunteer(**volunteerID**: integer, hoursVolunteered: integer)
- **Foreign key:** volunteerID is foreign key for AnimalCaretaker caretakerID

FundraiserEvent(eventID: integer, eventType: varchar, eventDate: date, eventTime: time, donationGoal: integer), eventDate is unique
- **Additional notes:** For every instance of FundraiserEvent, some AnimalCaretaker must have its eventID as a fundEventID, eventDate is unique since you shouldn't have more than one event on any given day.

FundraiserEventDate(eventDate: date, eventTime: time)

Post(postID: integer, description: varchar, postingDate: date **caretakerID**: integer), caretakerID not null, postingDate is unique
- **Foreign key:** creatorID is foreign key to animalCaretaker caretakerID
- **Additional notes:** postingDate is unique since you don't want to spam your follower's feeds

PostDateAndType(postingDate: date, postType: varchar)

Customer(customerID: integer, customerName: varchar)

Adopter(**adopterID**: integer, numOfAdoptions: integer, safeOwnerRating: integer, adopterPostalCode: varchar, adopterAddress: varchar)
- **Foreign key:** adopterID is foreign key for customerID in Customer

- **Additional notes:** Safe owner rating is on a scale of 1 to 5, determined by an employee

AdopterPCs(adopterPostalCode: varchar, adopterCity: varchar)

AdoptionDetails(adoptionID: integer, **petID**: integer, **adopterID**: integer, **caretakerID**: integer, adoptionDate: date, notes: varchar)

- **Foreign key:** petID is foreign key to petID in Animal
- **Foreign key:** adopterID is foreign key to adopterID in Adopter
- **Foreign key:** caretakerID is foreign key to caretakerID in AnimalCaretaker that facilitated adoption
- petID is unique since each pet can be adopted only once

PetAdopter(**petID**: integer, **adopterID**: integer)

- **Foreign key:** petID is foreign key to petID in Animal
- **Foreign key:** adopterID is foreign key to adopterID in Adopter

Appointment(**petID**: integer, **caretakerID**: integer, **customerID**: integer, apptDate: date, apptTime: time)

- **Foreign key:** petID is foreign key to petID in Animal
- **Foreign key:** caretakerID is foreign key to caretakerID in AnimalCaretaker that facilitates appointment
- **Foreign key:** customerID is foreign key to customerID in Customer

Donation(**customerID**: integer, **caretakerID**: integer, amount: integer)

- **Foreign key:** customerID is foreign key to customerID in Customer
- **Foreign key:** caretakerID is foreign key to caretakerID in AnimalCaretaker that facilitates donation

Item(itemID: integer, itemName: varchar, quantity: integer)

ItemPurchase(**customerID**: integer, **caretakerID**: integer, **itemID**: integer)

- **Foreign key:** customerID is foreign key to customerID in Customer
- **Foreign key:** caretakerID is a foreign key to caretakerID in AnimalCaretaker
- **Foreign key:** itemID is a foreign key to itemID in Items

ItemPrice(itemID: integer, total: integer)

6. SQL DDL Statements

The SQL DDL statements required to create all the tables from item #6. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc

```
CREATE TABLE Animal (
	petID INTEGER,
	animalName VARCHAR,
	type VARCHAR,
	age INTEGER,
	favouriteCaretaker INTEGER,
	previousOwner INTEGER,
	timeInShelter INTEGER,
	adopterID INTEGER,
	PRIMARY KEY (petID)
	FOREIGN KEY (favouriteCaretaker) REFERENCES AnimalCaretaker(caretakerID),
	FOREIGN KEY (previousOwner) REFERENCES Customer(customerID),
	FOREIGN KEY (adopterID) REFERENCES Adopter(adopterID);

CREATE TABLE VetAppointment (
	vetDate DATE,
	vetTime TIME,
	vetLicenseID INTEGER,
	reason VARCHAR,
	petID INTEGER,
	PRIMARY KEY (vetDate, vetTime),
	FOREIGN KEY (petID) REFERENCES Animal(petID));


CREATE TABLE Vet (
	vetLicenseID INTEGER,
	vetName VARCHAR
	PRIMARY KEY (vetLicenseID));

CREATE TABLE AnimalCaretakers (
	caretakerID INTEGER,
	caretakerName VARCHAR,
	fundEventID INTEGER,
	caretakerAddress VARCHAR,
	caretakerPostalCode VARCHAR,
	PRIMARY KEY (caretakerID)
	FOREIGN KEY (fundEventID) REFERENCES FundraiserEvent(eventID));
```

```
CREATE TABLE AnimalCaretakerPC (
        caretakerPostalCode VARCHAR,
        caretakerCity VARCHAR
        PRIMARY KEY (caretakerPostalCode));

CREATE TABLE Worker (
        workerID INTEGER,
        hourlyPay INTEGER,
        PRIMARY KEY (workerID)
        FOREIGN KEY (workerID) REFERENCES AnimalCaretaker(caretakerID));

CREATE TABLE Volunteer (
        volunteerID INTEGER,
        hoursVolunteered INTEGER,
        PRIMARY KEY (volunteerID),
        FOREIGN KEY (volunteerID) REFERENCES AnimalCaretaker(caretakerID));

CREATE TABLE FundraiserEvent (
        eventID INTEGER,
        eventType VARCHAR,
        eventDate DATE,
        eventTime TIME,
        donationGoal INTEGER,
        PRIMARY KEY (eventID),
        UNIQUE (eventDate));

CREATE TABLE FundraiserEventDate (
        eventDate DATE,
        eventTime TIME,
        PRIMARY KEY (eventDate),
        UNIQUE (eventDate));

CREATE TABLE Post (
        postID INTEGER,
        postType VARCHAR,
        description VARCHAR,
        postingDate DATE,
        caretakerID INTEGER,
        PRIMARY KEY (postID),
        FOREIGN KEY (caretakerID) REFERENCES AnimalCaretaker(caretakerID),
        UNIQUE(postingDate));

CREATE TABLE PostDateAndType (
        postingDate DATE,
        postType VARCHAR,
        PRIMARY KEY (postingDate),
        UNIQUE(postingDate));

CREATE TABLE Customer (
        customerID INTEGER,
        customerName VARCHAR,
        PRIMARY KEY (customerID));

CREATE TABLE Adopter (
        adopterID INTEGER,
        numOfAdoptions INTEGER,
        safeOwnerRating INTEGER,
        adopterPostalCode VARCHAR,
        adopterAddress VARCHAR,
        PRIMARY KEY (adopterID)
        FOREIGN KEY (adopterID) REFERENCES Customer(customerID));

CREATE TABLE AdopterPCs (
        adopterPostalCode VARCHAR,
        adopterCity VARCHAR,
        PRIMARY KEY (adopterPostalCode));

CREATE TABLE AdoptionDetails (
        adoptionID INTEGER,
        petID INTEGER UNIQUE,
        adopterID INTEGER,
        caretakerID INTEGER,
        adoptionDate DATE,
        notes VARCHAR,
        PRIMARY KEY (adoptionID)
        FOREIGN KEY (petID) REFERENCES Animal(petID),
        FOREIGN KEY (adopterID) REFERENCES Adopter(adopterID),
        FOREIGN KEY (caretakerID) REFERENCES AnimalCaretaker(caretakerID));
```

```
CREATE TABLE PetAdopter (
        petID INTEGER,
        adopterID INTEGER,
        PRIMARY KEY (petID)
        FOREIGN KEY (petID) REFERENCES Animal(petID),
        FOREIGN KEY (adopterID) REFERENCES Adopter(adopterID);

CREATE TABLE Appointment (
        petID INTEGER,
        caretakerID INTEGER,
        customerID INTEGER,
        apptDate DATE,
        apptTime TIME,
        PRIMARY KEY (petID, caretakerID, customerID),
        FOREIGN KEY (petID) REFERENCES Animal(petID),
        FOREIGN KEY (caretakerID) REFERENCES AnimalCaretaker(caretakerID),
        FOREIGN KEY (customerID) REFERENCES Customer(customerID));

CREATE TABLE Donation (
        customerID INTEGER,
        caretakerID INTEGER,
        amount INTEGER,
        PRIMARY KEY (customerID, caretakerID),
        FOREIGN KEY (customerID) REFERENCES Customer(customerID),
        FOREIGN KEY (caretakerID) REFERENCES AnimalCaretaker(caretakerID));

CREATE TABLE Item (
        itemID INTEGER,
        itemName VARCHAR,
        quantity INTEGER,
        PRIMARY KEY (itemID));

CREATE TABLE ItemPurchase (
        customerID INTEGER,
        caretakerID INTEGER,
        itemID INTEGER,
        PRIMARY KEY (customerID, caretakerID, itemID),
        FOREIGN KEY (customerID) REFERENCES Customer(customerID),
        FOREIGN KEY (caretakerID) REFERENCES AnimalCaretaker(caretakerID),
        FOREIGN KEY (itemID) REFERENCES Item(itemID));
```

```
CREATE TABLE ItemPrice (
        itemID INTEGER,
        total INTEGER,
        PRIMARY KEY (itemID);
```

7. Insert Statements

```
INSERT INTO Animal (petID, animalName, type, age, favouriteCaretaker, previousOwner,
timeInShelter, adopterID)
VALUES
        (1, 'Fluffy', 'Cat', 2, 3, 4, 12, 1),
        (2, 'Rex', 'Dog', 3, 1, 4, 10, 2),
        (3, 'Whiskers', 'Cat', 5, 2, 5, 8, 3),
        (4, 'Buddy', 'Bunny', 4, 4, 3, 9, 4),
        (5, 'Luna', 'Dog', 1, 3, 1, 11, 5),
        (6, 'Domino', 'Hamster', 1, 3, 1, 2, NULL),
        (7, 'Patch', 'Dog', 2, 5, 4, 2, NULL),
        (8, 'Pirate', 'Cat', 2, 4, 4, 11, NULL),
        (9, 'Cloudy', 'Bunny', 3, 2, NULL, 7, NULL),
        (10, 'Smoothie', 'Bunny', 3, 2, NULL, 7, NULL);

INSERT INTO VetAppointment (vetDate, vetTime, vetLicenseID, reason, petID)
VALUES
        ('2023-10-20', '14:00:00', 1, 'Checkup', 1),
        ('2023-10-21', '10:30:00', 2, 'Vaccination', 2),
        ('2023-10-22', '15:15:00', 3, 'Dental cleaning', 3),
        ('2023-10-23', '11:00:00', 4, 'Spaying', 4),
        ('2023-10-24', '13:45:00', 5, 'Checkup', 5);

INSERT INTO Vet (vetLicenseID, vetName)
VALUES
        (1, 'Dr. Allan'),
        (2, 'Dr. Papper'),
        (3, 'Dr. Lorde'),
        (4, 'Dr. Levette'),
        (5, 'Dr. Michaels');

INSERT INTO AnimalCaretakers (caretakerID, caretakerName, fundEventID,
caretakerAddress, caretakerPostalCode)
```

```
VALUES
        (1, 'John Peters', 1, '123 Main St', '12345'),
        (2, 'Mary Johnson', 2, '456 Elm St', '67890'),
        (3, 'David Perks', 3, '789 Oak St', '34567'),
        (4, 'Elaine Brown', 4, '101 Pine St', '87654'),
        (5, 'Michael Wilson', 5, '234 Maple St', '43210');

INSERT INTO AnimalCaretakerPC (caretakerPostalCode, caretakerCity)
VALUES
        ('12345', 'Narnia'),
        ('67890', 'Atlantis'),
        ('34567', 'Brokeburn'),
        ('87654', 'Lancaster'),
        ('43210', 'Columbus');

INSERT INTO Worker (workerID, hourlyPay)
VALUES
        (1, 15),
        (2, 17),
        (3, 14),
        (4, 18),
        (5, 16);

INSERT INTO Volunteer (volunteerID, hoursVolunteered)
VALUES
        (1, 20),
        (2, 25),
        (3, 18),
        (4, 30),
        (5, 22);

INSERT INTO FundraiserEvent (eventID, eventType, eventDate, eventTime, donationGoal)
VALUES
        (1, 'Charity Auction', '2023-11-01', '18:00:00', 5000),
        (2, 'Pet Walkathon', '2023-11-15', '10:00:00', 3000),
        (3, 'Adoption Fair', '2023-11-30', '14:30:00', 2000),
        (4, 'Pet Costume Contest', '2023-12-10', '15:00:00', 2500),
        (5, 'Animal Rescue Gala', '2023-12-25', '19:00:00', 7000);

INSERT INTO FundraiserEventDate (eventDate, eventTime)
```

```
VALUES
        ('2023-10-20', '18:00:00'),
        ('2023-10-21', '19:30:00'),
        ('2023-10-22', '17:45:00'),
        ('2023-10-23', '20:15:00'),
        ('2023-10-24', '09:00:00');

INSERT INTO Post (postID, postType, description, postingDate, caretakerID)
VALUES
        (1, 'Announcement', 'Adoption event this weekend!', '2023-10-25', 101),
        (2, 'News', 'New arrivals in the shelter', '2023-10-26', 102),
        (3, 'Update', 'Vet check-ups for all animals', '2023-10-27', 103),
        (4, 'Event', 'Volunteer appreciation day', '2023-10-28', 104),
        (5, 'Adoption', 'Adopt a furry friend today', '2023-10-29', 105);

INSERT INTO PostDateAndType (postingDate, postType)
VALUES
        ('2023-10-25', 'Announcement'),
        ('2023-10-26', 'News'),
        ('2023-10-27', 'Update'),
        ('2023-10-28', 'Event'),
        ('2023-10-29', 'Adoption');

INSERT INTO Customer (customerID, customerName)
VALUES
        (1, 'Alice Johnson'),
        (2, 'Bob Smith'),
        (3, 'Carol Davis'),
        (4, 'David Wilson'),
        (5, 'Eve Brown');

INSERT INTO Adopter (adopterID, numOfAdoptions, safeOwnerRating,
adopterPostalCode, adopterAddress)
VALUES
        (1, 2, 4, '12345', '123 Elm St'),
        (2, 0, 3, '23456', '456 Oak St'),
        (3, 1, 5, '34567', '789 Pine St'),
        (4, 3, 4, '45678', '101 Maple St'),
        (5, 2, 4, '56789', '234 Birch St');
```

```
INSERT INTO AdopterPCs (adopterPostalCode, adopterCity)
VALUES
        ('12345', 'Narnia'),
        ('23456', 'Atlantis'),
        ('34567', 'Brokeburn'),
        ('45678', 'Lancaster'),
        ('56789', 'Columbus');

INSERT INTO AdoptionDetails (adoptionID, petID, adopterID, caretakerID, adoptionDate,
notes)
VALUES
        (1, 1, 1, 1, '2023-10-20', 'friendly cat'),
        (2, 2, 2, 2, '2023-10-21', 'playful dog'),
        (3, 3, 3, 3, '2023-10-22', 'loud cat'),
        (4, 4, 4, 4, '2023-10-23', 'really soft bunny'),
        (5, 5, 5, 5, '2023-10-24', 'quiet dog');

INSERT INTO PetAdopter (petID, adopterID)
VALUES
        (1, 1),
        (2, 2),
        (3, 3),
        (4, 4),
        (5, 5);

INSERT INTO Donation (customerID, caretakerID, amount)
VALUES
        (1, 1, 100),
        (2, 2, 150),
        (3, 3, 200),
        (4, 4, 50),
        (5, 5, 75);

INSERT INTO Item (itemID, itemName, quantity)
VALUES
        (1, 'Pet Food', 100),
        (2, 'Blankets', 50),
        (3, 'Toys', 75),
        (4, 'Medicine', 25),
        (5, 'Leashes', 30);
```

```
INSERT INTO ItemPurchase (customerID, caretakerID, itemID)
VALUES
        (1, 1, 3),
        (2, 1, 5),
        (3, 3, 1),
        (4, 5, 4),
        (5, 3, 5);

INSERT INTO ItemPrice (itemID, total)
VALUES
        (5, 200),
        (2, 100),
        (1, 150),
        (4, 75),
        (3, 90);
```